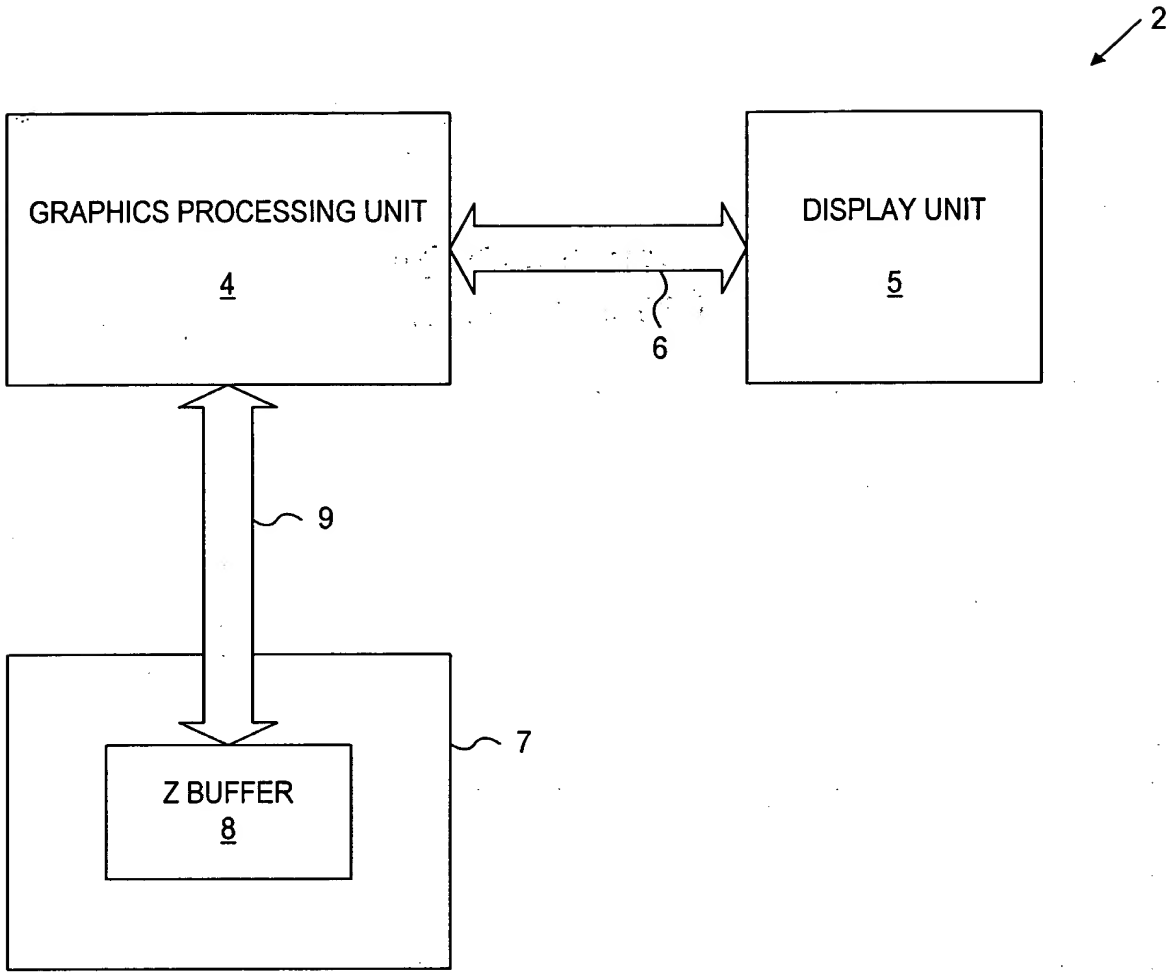

FIG. 1

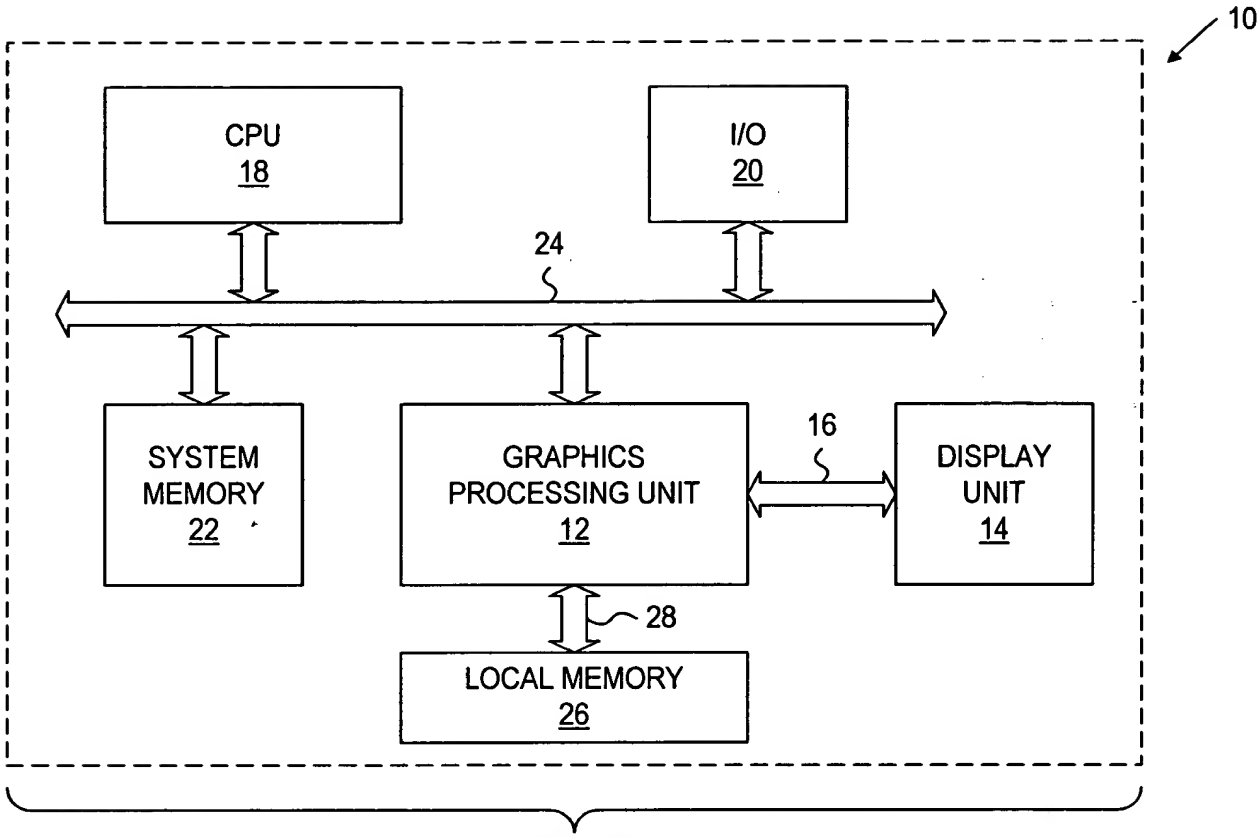
10
CPU
18
I/O
20
24
SYSTEM
MEMORY
22
GRAPHICS
PROCESSING UNIT
12
16
DISPLAY
UNIT
14
28
LOCAL MEMORY
26

FIG. 2A

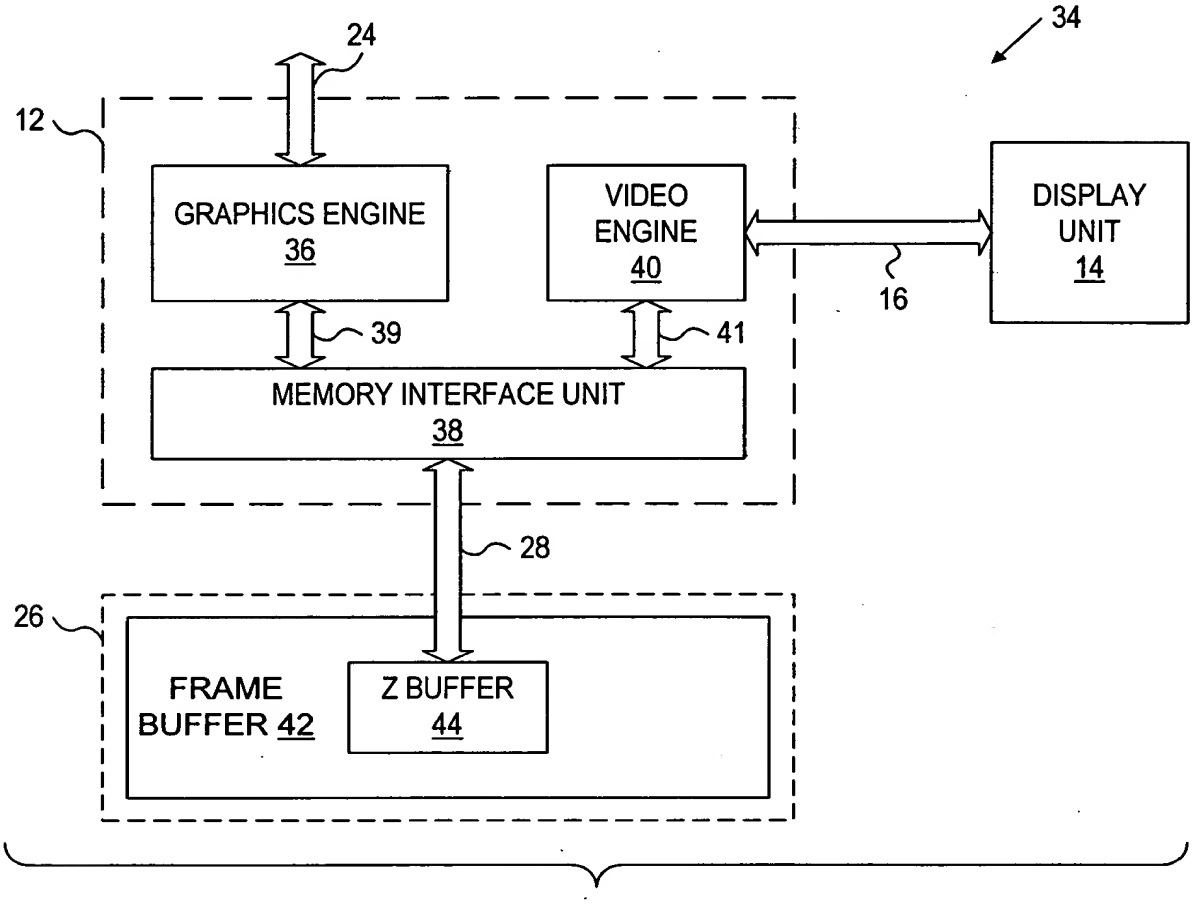
34
24
12
GRAPHICS ENGINE
36
VIDEO
ENGINE
40
DISPLAY
UNIT
14
16
39
41
MEMORY INTERFACE UNIT
38
28
26
FRAME
BUFFER 42
Z BUFFER
44

FIG. 2B

0828 0828960

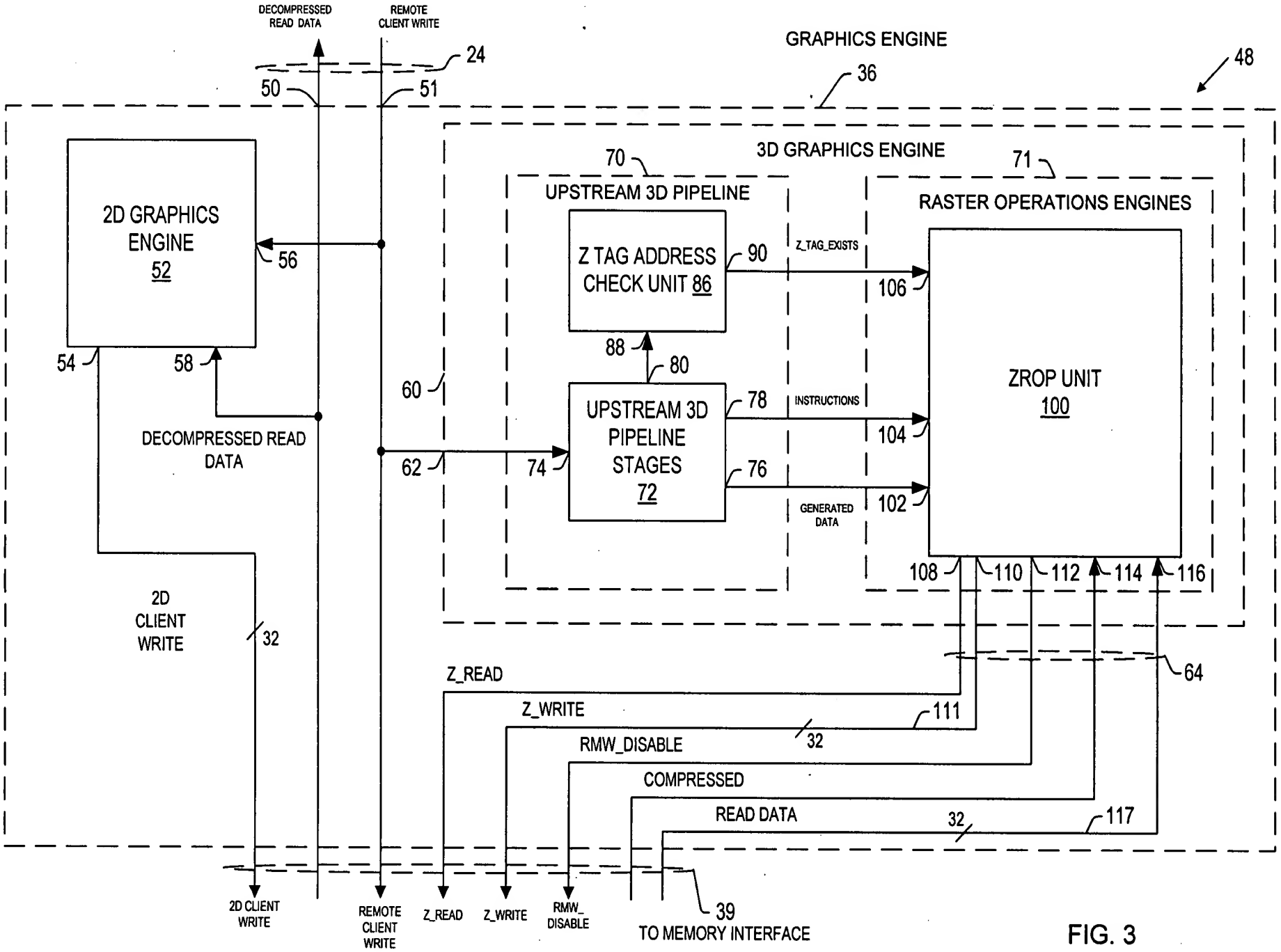

FIG. 3

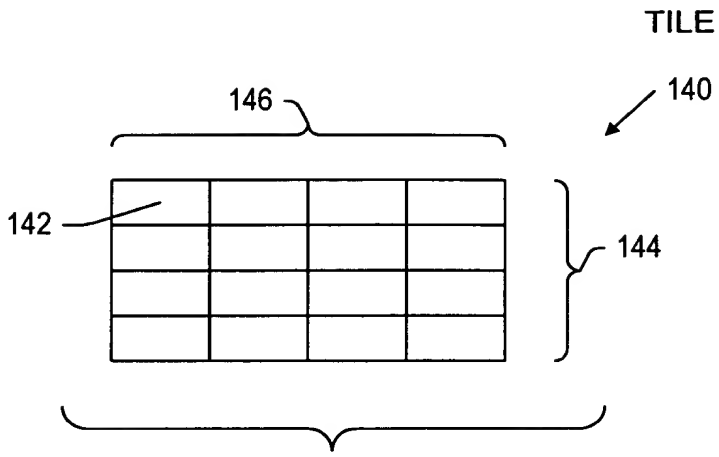

FIG. 4A

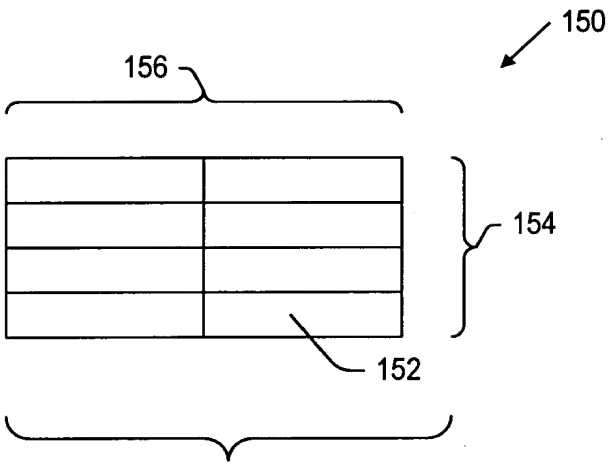

FIG. 4B

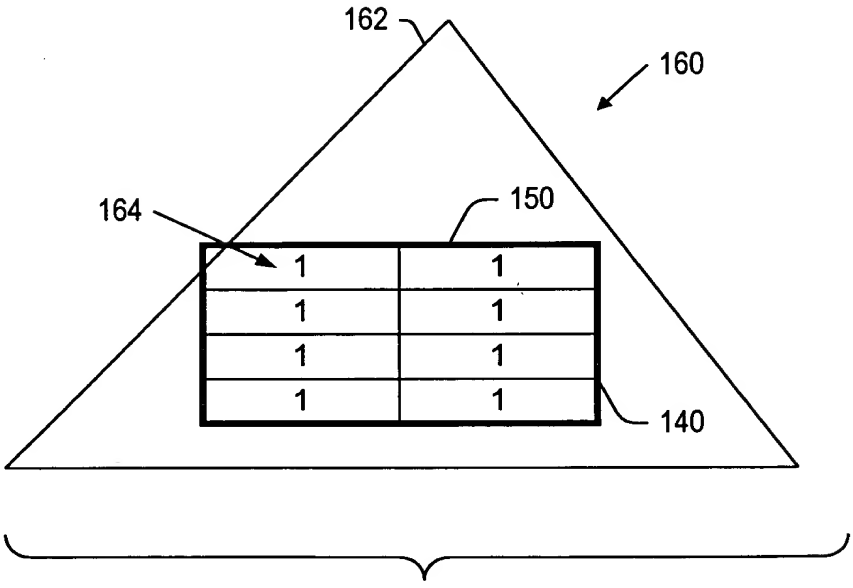

FIG. 4C

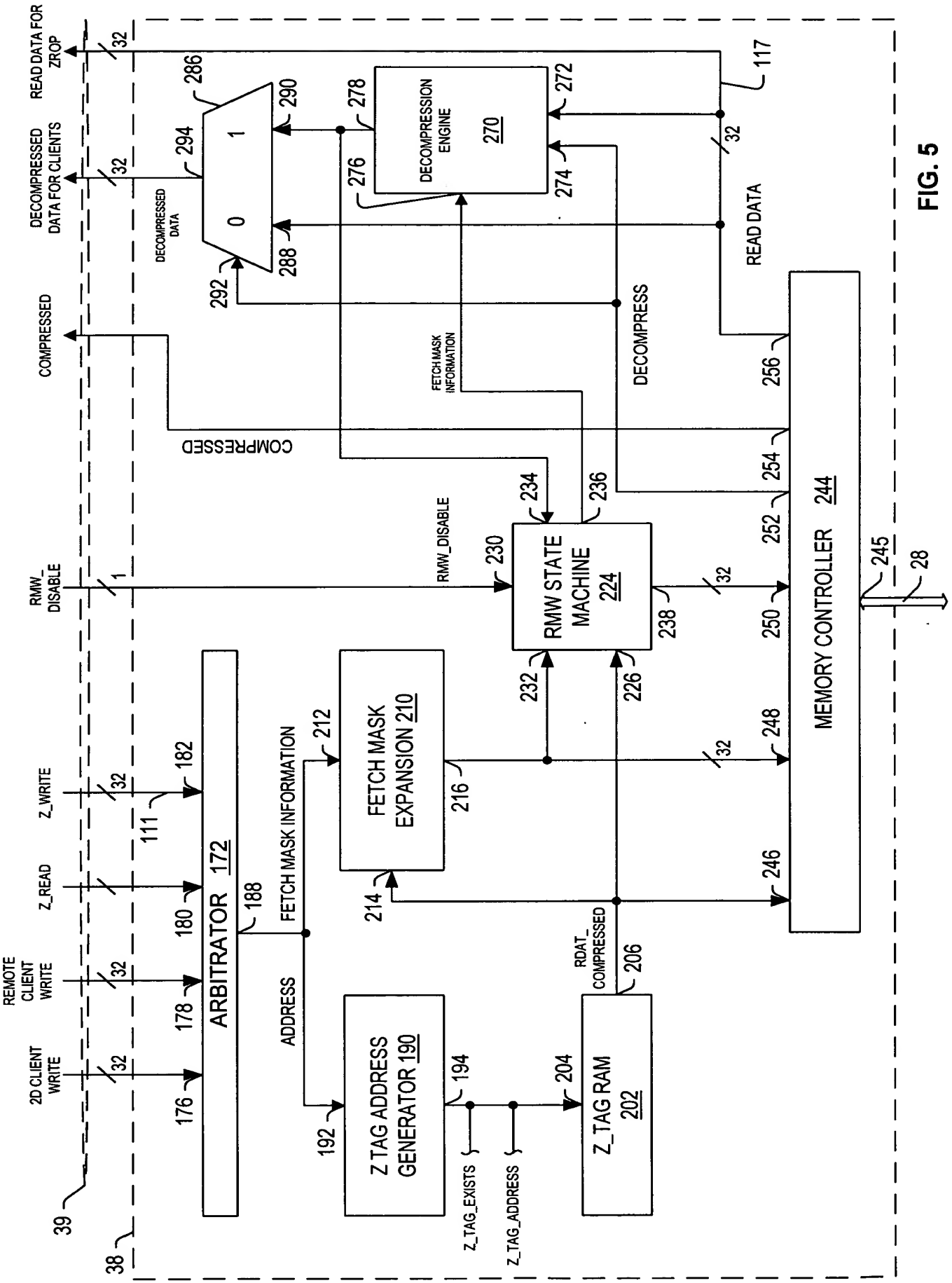
READ DATA FOR ZROP
DECOMPRESSED DATA FOR CLIENTS
COMPRESSED
RMW_ DISABLE
2D CLIENT WRITE
REMOTE CLIENT WRITE
Z_READ
Z_WRITE
ARBITRATOR 172
ADDRESS
FETCH MASK INFORMATION
Z TAG ADDRESS GENERATOR 190
Z_TAG_EXISTS
Z_TAG_ADDRESS
Z_TAG RAM 202
RDAT_ COMPRESSED
FETCH MASK EXPANSION 210
RMW_DISABLE
RMW STATE MACHINE 224
DECOMPRESSED DATA
DECOMPRESSION ENGINE 270
FETCH MASK INFORMATION
DECOMPRESS
READ DATA
MEMORY CONTROLLER 244
COMPRESSED
39
38
111
182
180
178
176
188
212
216
214
232
226
234
236
230
238
250
252
254
256
246
248
206
204
194
192
286
290
278
272
294
276
274
288
292
117
245
28

FIG. 5

0028AT 9E028960

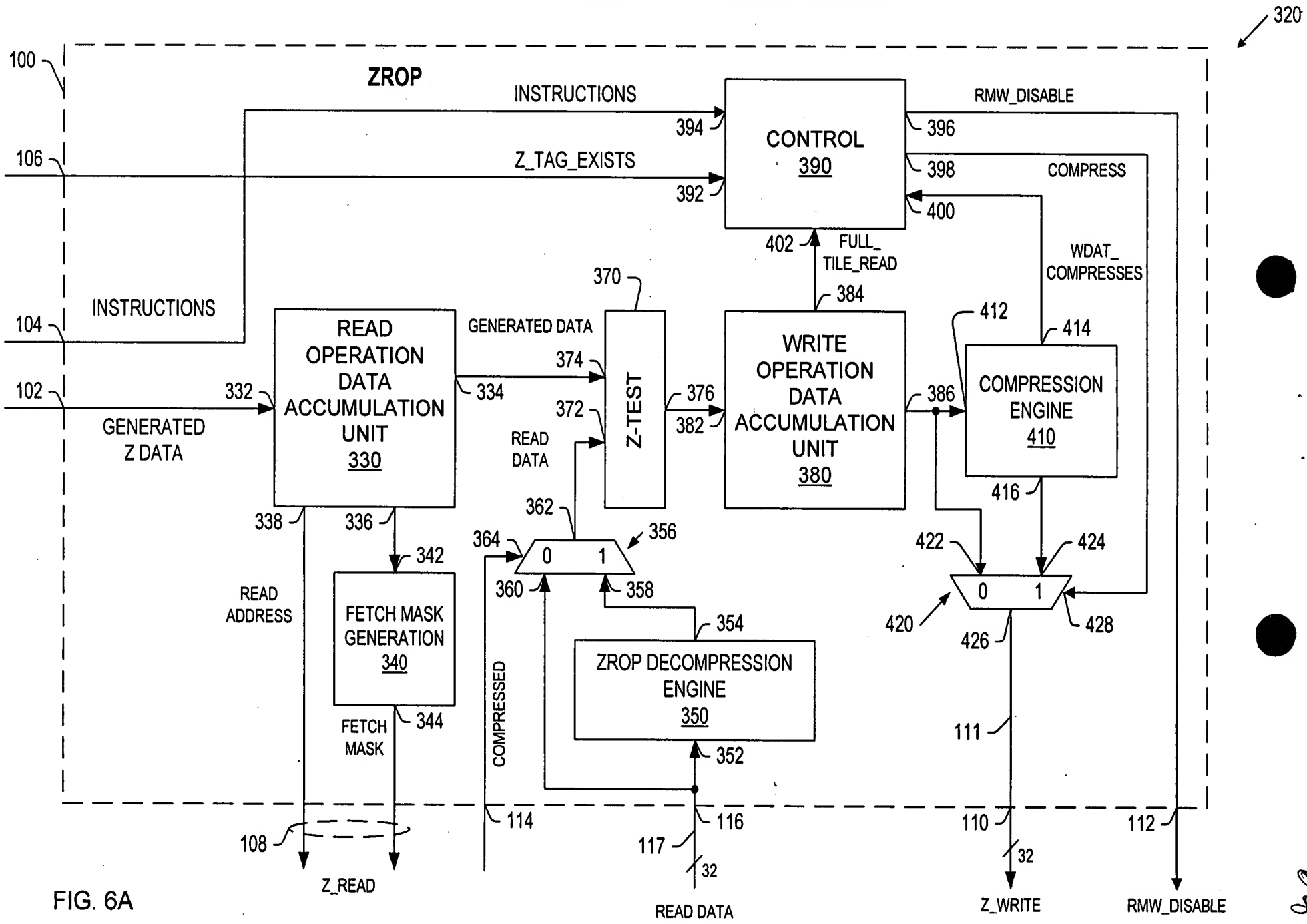

FIG. 6A

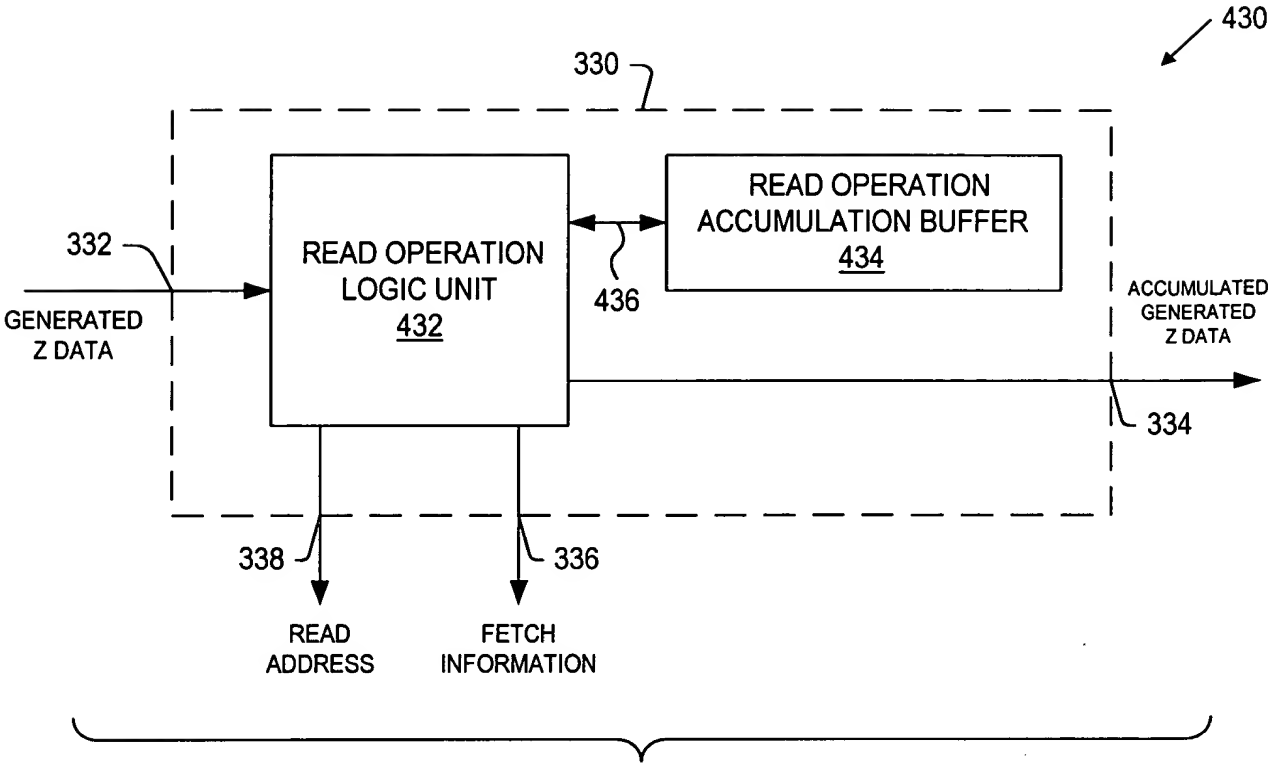
430
330
READ OPERATION LOGIC UNIT 432
READ OPERATION ACCUMULATION BUFFER 434
436
332
GENERATED Z DATA
ACCUMULATED GENERATED Z DATA
334
338
336
READ ADDRESS
FETCH INFORMATION

FIG. 6B

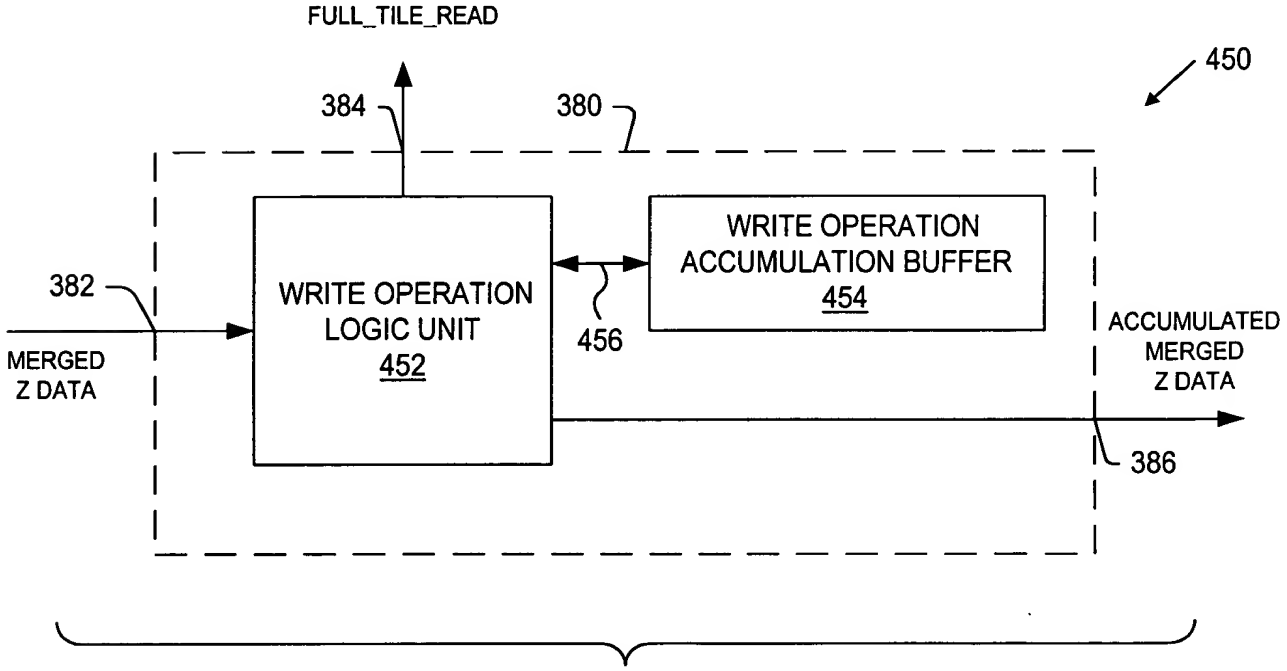
FULL_TILE_READ
450
384
380
WRITE OPERATION LOGIC UNIT 452
WRITE OPERATION ACCUMULATION BUFFER 454
456
382
MERGED Z DATA
ACCUMULATED MERGED Z DATA
386

FIG. 6C

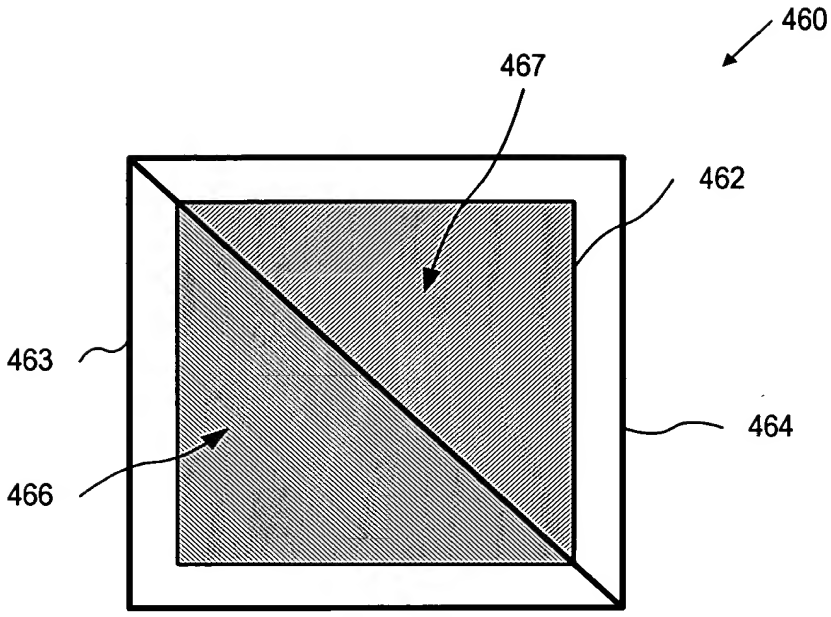

FIG. 6D

0050305-040200

470

WRITE WITH Z READ

Z_TAG_ EXISTS	RDAT_ COMPRESSED	WDAT_ COMPRESSES	FULL_ TILE_READ	ACTION		
				COMPRESS	FULL_ WRITE	RMW_ DISABLE
0	0	X	X	0	0	1
1	0	X	X	0	0	1
1	1	X	X	0	1	1
1	0	0	X	0	0	1
1	0	1	1	1	0	1
1	1	0	X	0	1	1
1	1	1	1	1	0	1

FIG. 7A

0606306 v1 4044000

472

WRITE WITHOUT Z READ

Z_TAG_ EXISTS	Z_COMP_ OK	RDAT_ COMPRESSED	WDAT_ COMPRESSES	FULL_ COVG	ACTION		
					COMPRESS	FULL_ WRITE	RMW_ DISABLE
0	0	-	X	X	0	0	1
1	0	-	X	0	0	0	0
1	0	-	X	1	0	0	1
1	1	-	0	0	0	0	0
1	1	-	0	1	0	0	1
1	1	-	1	1	1	0	1

FIG. 7B

474

FAST CLEAR

Z_TAG_ EXISTS	Z_COMP_ OK	RDAT_ COMPRESSED	WDAT_ COMPRESSES	FULL_ COVG	ACTION		
					COMPRESS	FULL_ WRITE	RMW_ DISABLE
0	0	-	1	X	0	0	1
1	0	-	1	0	0	0	1
1	0	-	1	1	0	0	1
1	1	-	1	0	1	0	1
1	1	-	1	1	1	0	1

FIG. 7C

05605.0036-00000

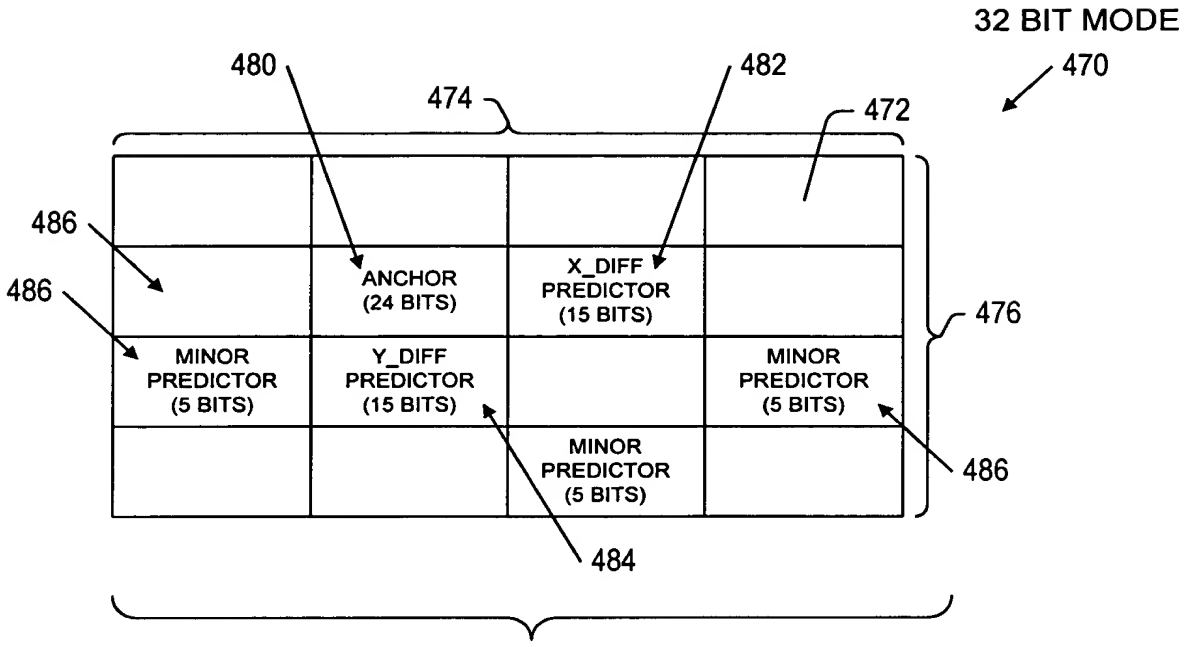
32 BIT MODE
470
480
482
474
472
486
486
ANCHOR
(24 BITS)
X_DIFF
PREDICTOR
(15 BITS)
476
MINOR
PREDICTOR
(5 BITS)
Y_DIFF
PREDICTOR
(15 BITS)
MINOR
PREDICTOR
(5 BITS)
MINOR
PREDICTOR
(5 BITS)
486
484

FIG. 8A

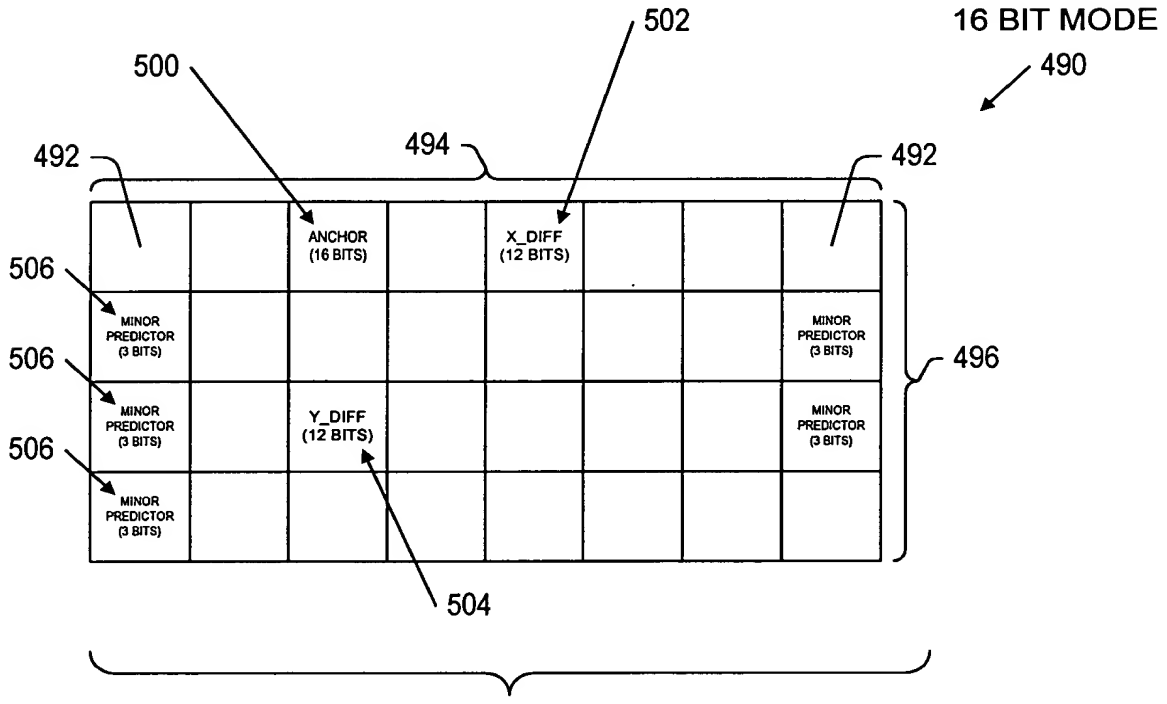
16 BIT MODE
490
500
502
494
492
492
ANCHOR
(16 BITS)
X_DIFF
(12 BITS)
506
MINOR
PREDICTOR
(3 BITS)
MINOR
PREDICTOR
(3 BITS)
496
506
MINOR
PREDICTOR
(3 BITS)
Y_DIFF
(12 BITS)
MINOR
PREDICTOR
(3 BITS)
506
MINOR
PREDICTOR
(3 BITS)
504

FIG. 8B

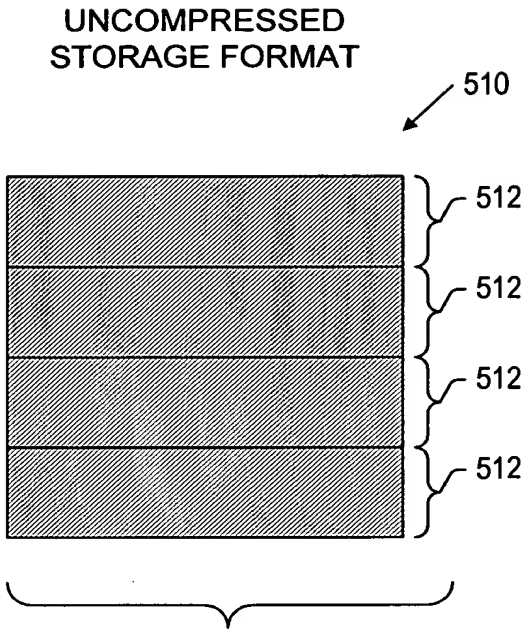

FIG. 8C

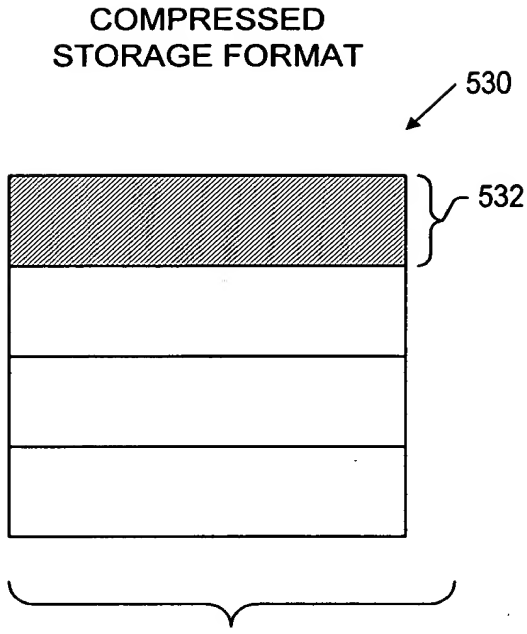

FIG. 8D

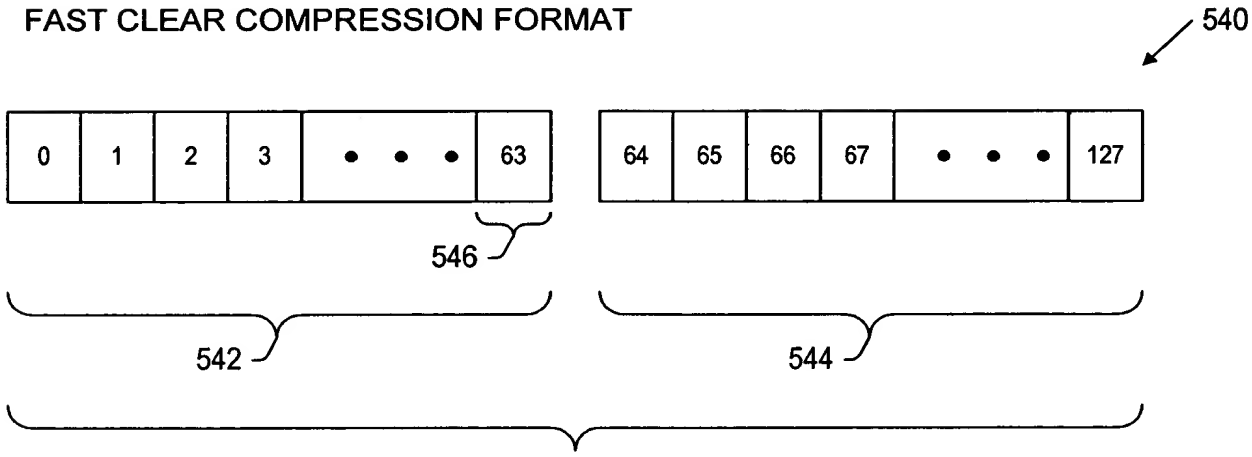

FIG. 8E

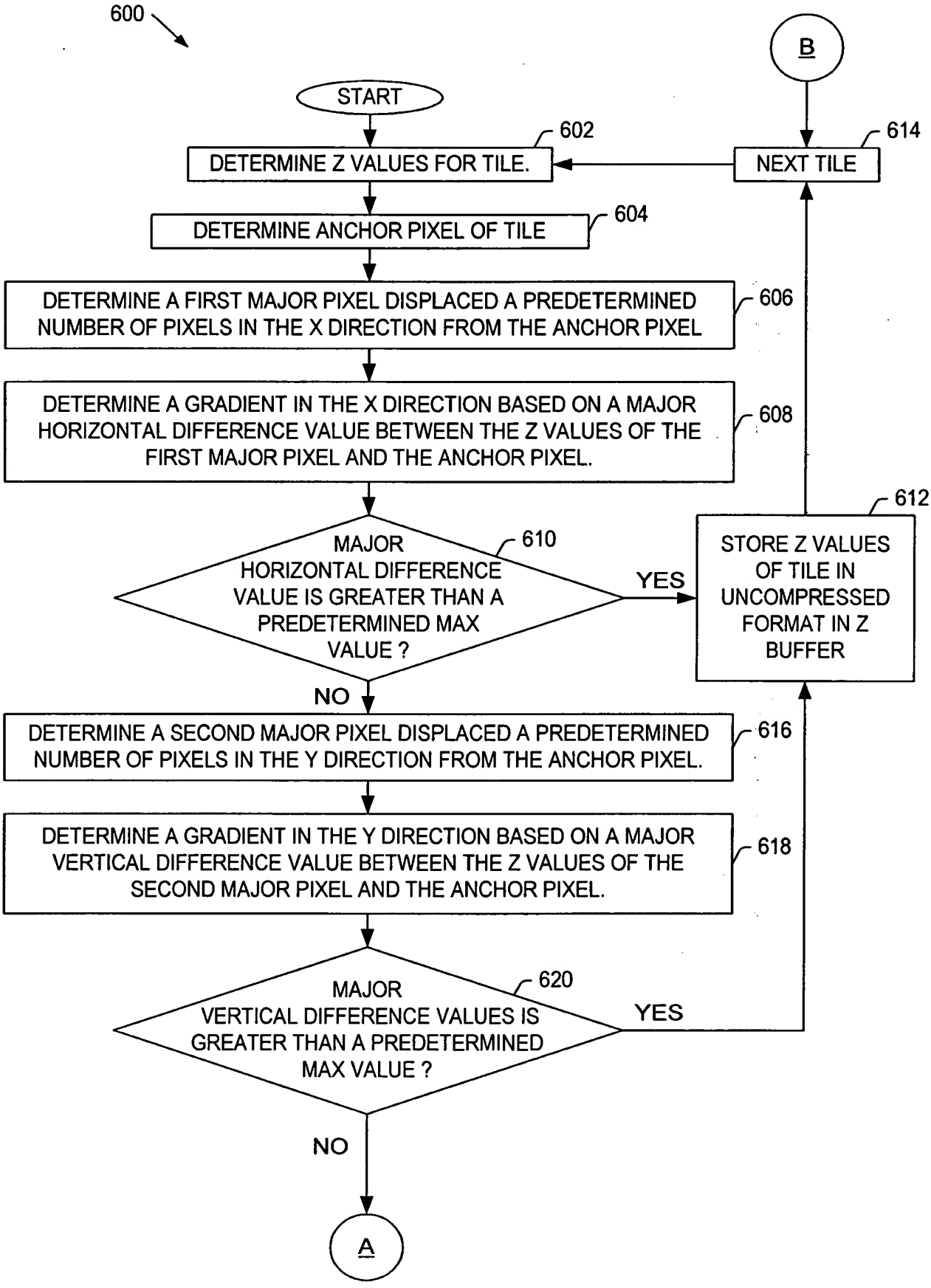
600
START
602
DETERMINE Z VALUES FOR TILE.
604
DETERMINE ANCHOR PIXEL OF TILE
606
DETERMINE A FIRST MAJOR PIXEL DISPLACED A PREDETERMINED NUMBER OF PIXELS IN THE X DIRECTION FROM THE ANCHOR PIXEL
608
DETERMINE A GRADIENT IN THE X DIRECTION BASED ON A MAJOR HORIZONTAL DIFFERENCE VALUE BETWEEN THE Z VALUES OF THE FIRST MAJOR PIXEL AND THE ANCHOR PIXEL.
610
MAJOR HORIZONTAL DIFFERENCE VALUE IS GREATER THAN A PREDETERMINED MAX VALUE ?
YES
612
STORE Z VALUES OF TILE IN UNCOMPRESSED FORMAT IN Z BUFFER
NO
616
DETERMINE A SECOND MAJOR PIXEL DISPLACED A PREDETERMINED NUMBER OF PIXELS IN THE Y DIRECTION FROM THE ANCHOR PIXEL.
618
DETERMINE A GRADIENT IN THE Y DIRECTION BASED ON A MAJOR VERTICAL DIFFERENCE VALUE BETWEEN THE Z VALUES OF THE SECOND MAJOR PIXEL AND THE ANCHOR PIXEL.
620
MAJOR VERTICAL DIFFERENCE VALUES IS GREATER THAN A PREDETERMINED MAX VALUE ?
YES
NO
A
B
614
NEXT TILE

FIG. 9A

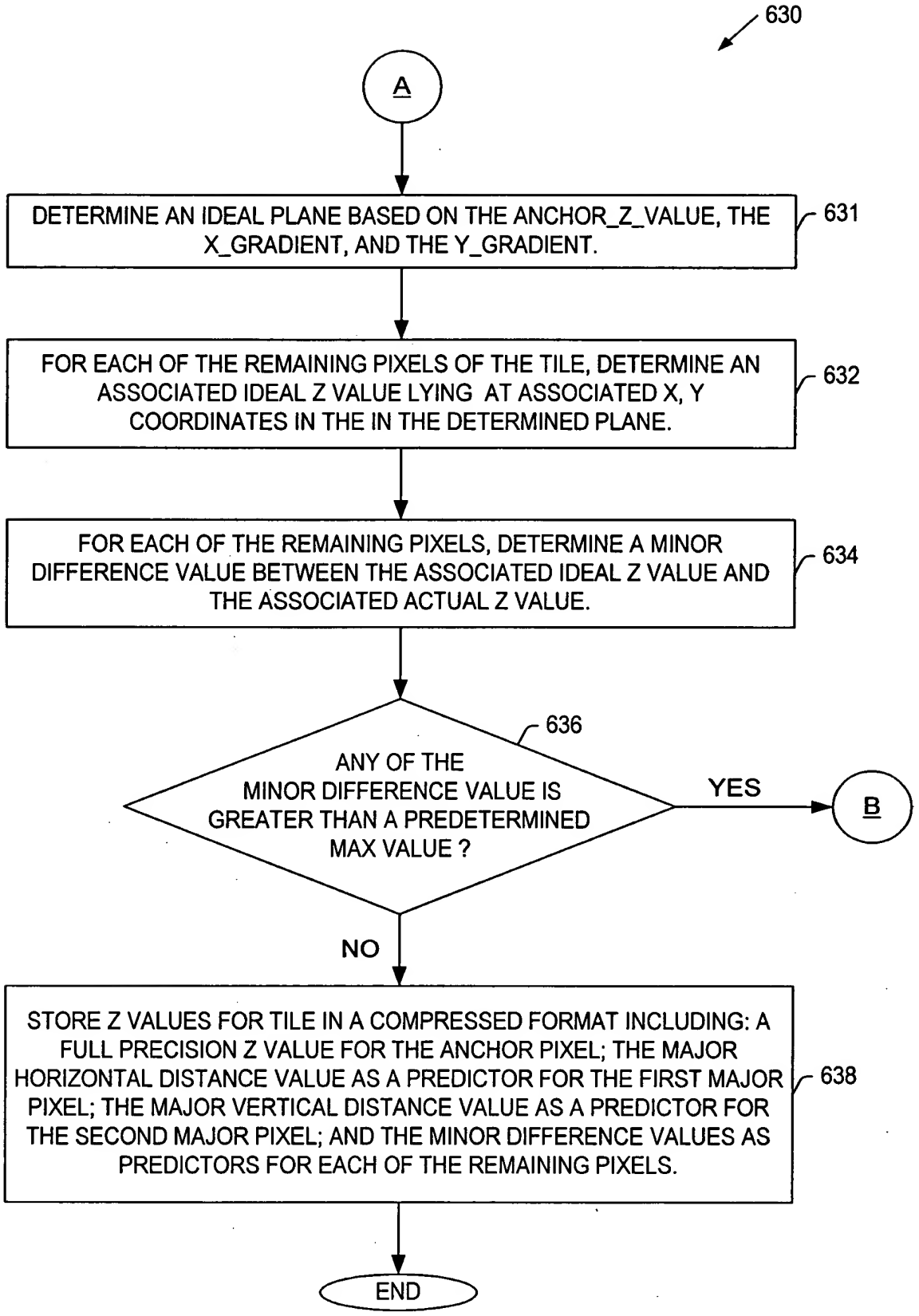
630
A
DETERMINE AN IDEAL PLANE BASED ON THE ANCHOR_Z_VALUE, THE X_GRADIENT, AND THE Y_GRADIENT.
631
FOR EACH OF THE REMAINING PIXELS OF THE TILE, DETERMINE AN ASSOCIATED IDEAL Z VALUE LYING AT ASSOCIATED X, Y COORDINATES IN THE IN THE DETERMINED PLANE.
632
FOR EACH OF THE REMAINING PIXELS, DETERMINE A MINOR DIFFERENCE VALUE BETWEEN THE ASSOCIATED IDEAL Z VALUE AND THE ASSOCIATED ACTUAL Z VALUE.
634
636
ANY OF THE MINOR DIFFERENCE VALUE IS GREATER THAN A PREDETERMINED MAX VALUE ?
YES
B
NO
STORE Z VALUES FOR TILE IN A COMPRESSED FORMAT INCLUDING: A FULL PRECISION Z VALUE FOR THE ANCHOR PIXEL; THE MAJOR HORIZONTAL DISTANCE VALUE AS A PREDICTOR FOR THE FIRST MAJOR PIXEL; THE MAJOR VERTICAL DISTANCE VALUE AS A PREDICTOR FOR THE SECOND MAJOR PIXEL; AND THE MINOR DIFFERENCE VALUES AS PREDICTORS FOR EACH OF THE REMAINING PIXELS.
638
END

FIG. 9B

650

Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Z8	Z9	Za	Zb	Zc	Zd	Ze	Zf
Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17
Z18	Z19	Z1a	Z1b	Z1c	Z1d	1e	Z1f

FIG. 10A

660

Z0	Z1	Z2	Z3
Z4	Z5	Z6	Z7
Z8	Z9	Za	Zb
Zc	Zd	Ze	Zf

FIG. 10B

670

Z2	anchor unsigned 16 bits
Z4_diff = Z4 – Z2	pred 12 bits
Z12_diff = Z12 - Z2	pred 12 bits
Z0_diff = Z0 – (Z2 – Z4_diff)	pred 3 bits
Z1_diff = Z1 – (Z2 – Z4_diff/2.0)	pred 3 bits
Z3_diff = Z3 – (Z2 + Z4_diff/2.0)	pred 3 bits
Z5_diff = Z5 – (Z2 + 3*Z4_diff/2.0)	pred 3 bits
Z6_diff = Z6 - (Z2 + 2*Z4_diff)	pred 3 bits
Z7_diff = Z7 - (Z2 + 5*Z4_diff/2.0)	pred 3 bits
Z8_diff = Z8 - (Z2 - Z4_diff + Z12_diff/2.0)	pred 3 bits
Z9_diff = Z9 - (Z2 - Z4_diff/2.0 + Z12_diff/2.0)	pred 3 bits
Za_diff = Za - (Z2 + Z12_diff/2.0)	pred 3 bits
Zb_diff = Zb - (Z2 + Z4_diff/2.0 + Zl2_diff/2.0)	pred 3 bits
Zc_diff = Zc - (Z2 + Z4_diff + Z12_diff/2.0)	pred 3 bits
Zd_diff = Zd - (Z2 + 3*Z4_diff/2.0 + Z12_diff/2.0)	pred 3 bits
Ze_diff = Ze - (Z2 + 2*Z4_diff + Z12_diff/2.0)	pred 3 bits
Zf_diff = Zf - (Z2 + 5*Z4_diff/2.0 + Z12_diff/2.0)	pred 3 bits
Z10_diff = Z10 - (Z2 - Z4_diff + Z12_diff)	pred 3 bits
Z11_diff = Z11 - (Z2 - Z4_diff/2.0 + Z12_diff)	pred 3 bits
Z13_diff = Z13 - (Z2 + Z4_diff/2.0 + Z12_diff)	pred 3 bits
Z14_diff = Z14 - (Z2 + Z4_diff + Z12_diff)	pred 3 bits
Z15_diff = Z15 - (Z2 + 3*Z4_diff/2.0 + Z12_diff)	pred 3 bits
Z16_diff = Z16 - (Z2 + 2*Z4_diff + Z12_diff)	pred 3 bits
Z17_diff = Z17 - (Z2 + 5*Z4_diff/2.0 + Z12_diff)	pred 3 bits
Z18_diff = Z18 - (Z2 - Z4_diff + 3*Z12_diff/2.0)	pred 3 bits
Z19_diff = Z19 - (Z2 - Z4_diff/2.0 + 3*Zl2_diff/2.0)	pred 3 bits
Zla_diff = Zla - (Z2 + 3*Zl2_diff/2.0)	pred 3 bits
Z1b_diff = Z1b - (Z2 + Z4_diff/2.0 + 3*Zl2_diff/2.0)	pred 3 bits
Z1c_diff = Z1c - (Z2 + Z4_diff + 3*Z12_diff/2.0)	pred 3 bits
Z1d_diff = Z1d - (Z2 + 3*Z4_diff/2.0 + 3*Zl2_diff/2.0)	pred 3 bits
Zle_diff = Zle - (Z2 + 2*Z4_diff + 3*Z12_diff/2.0)	pred 3 bits
Zlf_diff = Z1f - (Z2 + 5*Z4_diff/2.0 + 3*Zl2_diff/2.0)	pred 3 bits
fast_clear	1 bit

FIG. 11A

680

Compression format 16 bit Z:
bits [127:64] {Z1f_diff [2:0], Zle_diff [2:0], Z1d_diff [2:0], Z1c_diff [2:0], Z1b_diff [2:0], Zla_diff [2:0], Z19_diff [2:0], Z18_diff [2:0], Z17_diff [2:0], Z16_diff [2:0], Z15_diff [2:0], Z14_diff [2:0], Z13_diff [2:0], Z11_diff [2:0], Z10_diff [2:0], Zf_diff [2:0], Ze_diff [2:0], Zd_diff [2: 0], Zc_diff [2:0], Zb_diff [2:0], Za_diff [2:0], Z9_diff [2]}
bits [63:0] = {fast_clear, Z9_diff [1:0], Z8_diff[2:0], Z7_diff[2:0], Z6_diff[2:0], Z5_diff [2:0], Z3_diff [2:0], Z1 _diff [2:0], Z0_diff [2:0], Z12_diff [11:0], Z4_diff [11:0], Z2 [15:0]}
When cleared: bits [63:0] = (1'b1, 47'b0, Z_clear_value[15:0]}

FIG. 11B

0056036.4250

700

Z5	anchor unsigned 24 bits
Z6_diff = Z6 - Z5	pred 15 bits
Z9_diff = Z9 - Z5	pred 15 bits
Z0_diff = Z0 - (Z5 - Z9_diff - Z6_diff)	pred 5 bits
Z1_diff = Z1 - (Z5 - Z9_diff)	pred 5 bits
Z2_diff = Z2 - (Z5 - Z9_diff + Z6_diff)	pred 5 bits
Z3_diff = Z3 - (Z5 + 2*Z6_diff - Z9_diff)	pred 5 bits
Z4_diff = Z4 - (Z5 - Z6_diff)	pred 5 bits
Z7_diff = Z7 - (Z5 + 2*Z6_diff)	pred 5 bits
Z8_diff = Z8 - (Z5 + Z9_diff - Z6_diff)	pred 5 bits
Za_diff = Za - (Z5 + Z9_diff + Z6_diff)	pred 5 bits
Zb_diff = Zb - (Z5 + Z9_diff + 2*Z6_diff)	pred 5 bits
Zc_diff = Zc - (Z5 + 2*Z9_diff - Z6_diff)	pred 5 bits
Zd_diff = Zd - (Z5 + 2*Z9_diff)	pred 5 bits
Ze_diff = Ze - (Z5 + 2*Z9_diff + Z6_diff)	pred 5 bits
Zf_diff = Zf - (Z5 + 2*Z9_diff + 2*Z6_diff)	pred 5 bits
fast_clear	1 bit

FIG. 12A

190819

710

Compression format 24 bit Z/8 bit stencil:
bits [127:64] = {Zf_diff [4:0], Ze_diff [4:0], Zd_diff [4:0], Zc_diff [4:0], Zb_diff [4:0], Za_diff [4:0], Z8_diff [4:0], Z7_diff [4:0], Z4_diff [4:0], Z3_diff [4:0], Z2_diff [4:0], Z1_diff [4:0], Z0_diff [4:1]}
bits [63:0] = {fast_clear, Z0_diff [0], Z9_diff[14:0], Z6_diff[14:0], Z5[23:0], stencil[7:0]}
When cleared: bits [63:0] = {1'b1, 31'b0, Z_clear_value[23:0], stencil_clear_value[7:0]}

FIG. 12B